Amendments to the Claims:

This listing of claims will replace all prior versions, and listings of claims in the application:

Listing of Claims:

1. (Original) A method for use by a first process executing in a computer system for interacting with a second process executing in the computer system, the method comprising:

during a startup sequence of the second process, creating a copy of a global notification hook of the first process in the second process;

using the copy of the global notification hook, detecting an occurrence of a triggering message passed between an operating system and a thread of the second process;

in response to detecting the occurrence of the triggering message, determining whether subsequent messages passed between the operating system and the thread of the second process should be monitored; and

in the event that the subsequent messages should be monitored, activating a thread-level message hook within the thread of the second process, wherein the thread-level message hook is configured to monitor the subsequent messages.

2. (Original) The method of claim 1, wherein the thread-level message hook is further configured to cause an action to occur in response to a specified subsequent message.

3. (Original) The method of claim 2, wherein the action includes creating a visual effect for a window of the second process.

4. (Original) The method of claim 1, wherein the thread-level message hook is configured so as not to affect operation of a third process executing concurrently with the second process in the computer system.

5. (Original) The method of claim 1, wherein the triggering message is a window creation message.

6. (Original) The method of claim 5, wherein the act of determining whether subsequent messages should be monitored includes determining whether the window creation message relates to a window of interest.

7. (Original) The method of claim 6, wherein the window creation message relates to a window of interest unless one or more of the following conditions obtains: (a) the window creation message does not relate to a visible window; (b) the created window has a window type designated by a user as not being of interest; and (c) the created window has a window type that is incompatible with the thread-level message hook.

8. (Original) The method of claim 1, wherein the first process is a desktop management process.

9. (Original) The method of claim 8, wherein the second process is an application process.

10. (Original) The method of claim 1, further comprising, during a startup sequence of the first process:

detecting a third process executing in the computer system;

inserting a copy of the global notification hook into the third process; and

broadcasting a private startup message to the copy of the global notification hook in the third process.

11. (Original) The method of claim 10, wherein, in response to the private startup message, the copy of the global notification hook executes acts of:

determining whether subsequent messages passed between the operating system and a thread of the third process should be monitored; and

in the event that the subsequent messages should be monitored, activating a thread-level message hook within the thread of the third process.

12. (Original) The method of claim 11, wherein determining whether subsequent messages passed between the operating system and the thread of the third process should be monitored includes:

identifying a previously created window of the third process; and

determining whether the previously created window is of interest, wherein subsequent messages should be monitored in the event that the previously created window is of interest.

13. (Original) The method of claim 1, wherein the act of activating the thread-level message hook includes mapping executable code for the thread-level message hook into an address space of the second process.

14. (Original) The method of claim 1, wherein the act of creating the copy of the global notification hook includes mapping executable code for the global notification hook into an address space of the second process.

15. (Original) The method of claim 1, wherein the act of detecting the occurrence of the triggering message includes receiving message data of the triggering message.

16. (Original) The method of claim 15, wherein the message data of the triggering message is provided to the copy of the global notification hook concurrently with a transmission of the triggering message to the thread of the second process.

17. (Original) The method of claim 16, wherein the second process has a process-specific message queue that receives the transmitted message data of the triggering message.

18. (Original) A method for use by a first process executing in a computer system for interacting with a second process executing in the computer system, the method comprising:

during a startup sequence of the first process, creating a copy of a global notification hook of the first process in the second process; and

broadcasting a private startup message from the first process to the copy of the global notification hook;

wherein, in response to the private startup message, the copy of the global notification hook executes acts of:

determining whether subsequent messages passed between the operating system and a thread of the second process should be monitored; and

in the event that subsequent messages should be monitored, activating a thread-level message hook within the thread of the second process, wherein the thread-level message hook is configured to monitor the subsequent messages.

19. (Original) The method of claim 18, wherein the thread-level message hook is configured so as not to affect operation of a third process executing concurrently with the second process in the computer system.

20. (Original) The method of claim 18, wherein the thread-level message hook is further configured to cause an action to occur in response to a specified subsequent message.

21. (Original) The method of claim 18, wherein the action includes creating a visual effect for a window of the second process.

22. (Original) The method of claim 18, wherein the global notification hook determines that subsequent messages should be monitored in the event that a window of interest exists in the second process.

23. (Original) The method of claim 22, wherein a window existing in the second process is of interest unless one or more of the following conditions obtains: (a) the window is not a visible window; (b) the window has a window type designated by a user as not being of interest; and (c) the window has a window type that is incompatible with the thread-level message hook.

24. (Original) The method of claim 18, wherein the first process is a desktop management process.

25. (Original) The method of claim 24, wherein the second process is an application process.

26. (Original) The method of claim 18, wherein the act of activating the thread-level message hook includes mapping executable code for the thread-level message hook into an address space of the second process.

27. (Currently Amended) A computer program product stored on a computer-readable medium for use by a first process executing in a computer system for interacting with a second process executing in the computer system, the computer program product comprising:

~~a computer readable medium encoded with program code for a global notification hook of a first process, wherein the program code for the global notification hook is adapted to be copied into a second process during a startup sequence of the send process, the program code for the global notification hook including:~~

program code for creating a copy of a global notification hook of the first process in the second process during a startup sequence of the second process;

program code for detecting an occurrence of a triggering message in the second process;

program code for determining, in response to detecting the occurrence of the triggering message, whether subsequent messages passed between the operating system and a thread of the second process should be monitored; and

program code for activating a thread-level message hook within the thread of the second process in the event that the subsequent messages should be monitored, wherein the thread-level message hook is configured to monitor the subsequent messages.

28. (Currently Amended) The computer program product of claim 27, wherein the thread-level message hook is further configured to cause an action to occur in response to a specified subsequent message. ~~wherein the computer readable medium is further encoded with program code for the thread-level message hook, the program code for the thread-level message hook including:~~

~~program code for controlling an action to be taken in the event that a selected subsequent message is detected.~~

29. (Currently Amended) The computer program product of claim 27, wherein the computer readable medium comprises a magnetic storage medium ~~encoded with the program code~~.

30. (Currently Amended) The computer program product of claim 27, wherein the computer readable medium comprises an optical storage medium ~~encoded with the program code~~.

31. (Canceled)

32. (Currently Amended) A computer program product stored on a computer-readable medium for use by a first process executing in a computer system for interacting with a second process executing in the computer system, the computer program product comprising:

program code for creating a copy of a global notification hook of the first process in the second process during a startup sequence of the second process;

~~a computer readable medium encoded with program code for a global notification hook of a first process wherein the program code for the global notification hook is adapted to be copied into a second process during a startup sequence of the first process, the program code for the global notification hook including:~~

program code for broadcasting a private startup message from the first process to the copy of the global notification hook;

program code for determining, in response to the private startup message, whether subsequent messages passed between the operating system and a thread of the second process should be monitored; and

program code for activating a thread-level message hook within the thread of the second process, in the event that the subsequent messages should be monitored, wherein the thread-level message hook is configured to monitor the subsequent messages.

33. (Currently Amended) The computer program product of claim 32, wherein the thread-level message hook is further configured to cause an action to occur in response to a specified subsequent message. ~~wherein the computer readable medium is further encoded with program code for the thread-level message hook, the program code for the thread-level message hook including:~~

~~program code for controlling an action to be taken in the event that a specified subsequent message is detected.~~

34. (Currently Amended) The computer program product of claim 32, wherein the computer readable medium comprises a magnetic storage medium ~~encoded with the program code~~.

35. (Currently Amended) The computer program product of claim 32, wherein the computer readable medium comprises an optical storage medium ~~encoded with the program code~~.

36. (Canceled)